Government Smart Card Interoperability Specification v2.1 (NISTIR 6887 - 2003 Edition) Basic Services Interface C Language Binding

Conformance Test Assertions

FINAL DRAFT

Alan Goldfine

January 21, 2005

This document contains the conformance test assertions for the functions comprising the C language binding of the Basic Services Interface of version 2.1 of the Government Smart Card Interoperability Specification (GSC-IS), as contained in NIST Interagency Report 6887, 2003 Edition.

The 23 sections of this document correspond to the 23 C functions specified in Appendix E of the GSC-IS, which in turn correspond to the 23 BSI functions specified in Chapter 4 of the BSI. The test assertions for each of the functions, except gscBsiUtilGetExtendedErrorText() (q.v.), are numbered in the form X.Y, where X is the function number, and Y is the number of the assertion for that function. Thus, 7.6 is the number given to assertion 6 for function 7 (gscBsiUtilGetCardProperties()).

Variable names have the form Zxy. Here, Z is the name of the variable, and x and y are the two-digit representations of X and Y, where X.Y is the test assertion where the variable is defined. x and y have leading zeroes, if necessary. For example, unUniqueIDLen0706 is one of the variables defined in assertion 6 for function 7. Variables that are defined in the "Starting state for each Assertion", and are therefore global across all the assertions for the given function, have y = 00.

Conformance test assertions are statements of behavior, action, or condition that can be measured or tested. The assertions here are derived from the GSC-IS, and bridge the gap between the narrative of the GSC-IS and the test cases that comprise the BSI C language binding conformance test suite. Each test assertion is an independent, complete, and testable statement, and may result in one or more test cases.

1. gscBsiUtilAcquireContext()

```
unsigned          long                    gscBsiUtilAcquireContext(
    IN            UTILCardHandle                hCard
    IN            unsigned char *               uszAID
    IN            unsigned long                 unAIDLen
    IN            BSIAuthenticator *            strctAuthenticator
    IN            unsigned long                 unAuthNb
)
```

```
struct BSIAuthenticator is
       unsigned long          unAccessMethodType
       unsigned long          unKeyIDOrReference
       unsigned char          uszAuthValue[BSI_AUTHENTICATOR_MAX_LEN]
       unsigned long          unAuthValueLen
   where BSI_AUTHENTICATOR_MAX_LEN is implementor defined.
```

References:

1. GSC-IS 4.5.3.
2. GSC-IS E.4.1.

Starting state for each Assertion:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0100.
2. The card has a target container.
3. There exists an array strctAuthenticator0100, containing the authenticators associated with the target container.
4. There is no authenticated session established with any container on the connected card

Assertion 1.1

Purpose: To test gscBsiUtilAcquireContext() using valid parameters.

Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:

1. The call returns
   - the return code BSI_OK or the return code BSI_TERMINAL_AUTH.
2. An authenticated session is established with the target container.

Assertion 1.2

Purpose: To test gscBsiUtilAcquireContext() using a bad handle.

Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard /= hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of the string representing the AID value of the target container
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE.

2. An authenticated session is not established with the target container.

Assertion 9.1.2.1

Assertion 9.1.2.2

(See section 9.)

Assertion 1.3

Purpose: To test gscBsiUtilAcquireContext() with a bad AID value.

Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID.

2. An authenticated session is not established with the target container.

Assertion 9.1.3.1

Assertion 9.1.3.2

(See section 9.)

Assertion 1.4

Purpose: To test gscBsiUtilAcquireContext() with a bad AID length.

Scenario:

1. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM.

2. An authenticated session is not established with the target container.

Assertion 9.1.4.1

Assertion 9.1.4.2

(See section 9.)

Assertion 1.5

Purpose: To test gscBsiUtilAcquireContext() using an extraneous authenticator.

Scenario:

1. strctAuthenticator0105, an array of authenticators, at least one of which is not associated with the target container, is declared.

2. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0105
   - unAuthNb == the number of authenticators in strctAuthenticator0105.

Expected Results:

1. The call returns
   - the return code BSI_ACR_NOT_AVAILABLE.

2. An authenticated session is not established with the target container.

Assertion 9.1.5.1

Assertion 9.1.5.2

(See section 9.)

Assertion 1.6

Purpose: To test gscBsiUtilAcquireContext() using a bad authenticator.

Scenario:

1. strctAuthenticator0106, an array of authenticators, at least one of which is invalid, is declared.

2. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0106
   - unAuthNb == the number of authenticators in strctAuthenticator0106.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AUTH.

2. An authenticated session is not established with the target container.

Assertion 9.1.6.1

Assertion 9.1.6.2

(See section 9.)

Assertion 1.7

Purpose: To test gscBsiUtilAcquireContext() with a removed card.

Scenario:

1. The connected card is removed from the reader.

2. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 1.8

Purpose: To test gscBsiUtilAcquireContext() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.

3. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED.

Assertion 9.1.8.1

Assertion 9.1.8.2

(See section 9.)

Assertion 1.9

Purpose: To test gscBsiUtilAcquireContext() with a blocked PIN.

Scenario:
1. The PIN for the container whose AID value is uszAID0100 is blocked.

2. A gscBsiUtilAcquireContext() call is made to the SPS, with
   - hCard == hCard0100
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctAuthenticator == strctAuthenticator0100
   - unAuthNb == the number of authenticators in strctAuthenticator0100.

Expected Results:
1. The call returns
   - the return code BSI_PIN_BLOCKED.

2. An authenticated session is not established with the target container on the connected card.

Assertion 9.1.9.1

Assertion 9.1.9.2

(See section 9.)

2. gscBsiUtilConnect()

```
unsigned    long                    gscBsiUtilConnect(
    IN      unsigned char *             uszReaderName
    IN      unsigned long               unReaderNameLen
    OUT     UTILCardHandle *            hCard
)
```

References:
1. GSC-IS 4.5.4.
2. GSC-IS E.4.2.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a particular reader available to the candidate SPS.

2. The card is not connected to the reader.

3. The name of the reader is represented by a string uszReaderName0200.

4. There is a pointer hCard0200 to an unsigned long variable.

Assertion 2.1

Purpose: To test gscBsiUtilConnect() using a good card inserted into a specified reader.

Scenario:
1. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == uszReaderName0200
   - unReaderNameLen == the length of uszReaderName
   - hCard == hCard0200.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - hCard0200 == a valid handle.

2. The card is connected, with handle pointed to by hCard0200, to the reader uszReaderName0200.

Assertion 2.2

Purpose: To test gscBsiUtilConnect() using a good card inserted into a non-specified reader.

Scenario:
1. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == NULL
   - unReaderNameLen == 0
   - hCard = hCard0200.

Expected Results:
1. The call returns

- the return code BSI_OK
- hCard0200 == a valid handle.

The card is connected, with handle pointed to by hCard0200, to the first available reader.

Assertion 2.3

Purpose: To test gscBsiUtilConnect() using a bad reader name length.

Scenario:
1. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == uszReaderName0200
   - unReaderNameLen /= the length of uszReaderName
   - hCard == hCard0200.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM.

Note: There are no gscBsiUtilGetExtendedErrorText() assertions following assertion 2.3 because the card is presumably not connected.

Assertion 2.4

Purpose: To test gscBsiUtilConnect() using a bad reader name.

Scenario:
1. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == a string that does not represent a valid reader
   - unReaderNameLen == the length of uszReaderName
   - hCard == hCard0200.

Expected Results:
1. The call returns
   - the return code BSI_UNKNOWN_READER.

Note: There are no gscBsiUtilGetExtendedErrorText() assertions following assertion 2.4 because the card is presumably not connected.

Assertion 2.5

Purpose: To test gscBsiUtilConnect() with no card in the reader.

Scenario:
1. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == uszReaderName0200
   - unReaderNameLen == the length of uszReaderName
   - hCard == hCard0200.

Expected Results:
1. The call returns
   - the return code BSI_CARD_ABSENT.

Note: There are no gscBsiUtilGetExtendedErrorText() assertions following assertion 2.5 because the card is presumably not connected.

Assertion 2.6

Purpose: To test gscBsiUtilConnect() using a bad inserted card.

Scenario:

1. A bad card is in a particular reader available to the candidate SPS.

2. A gscBsiUtilConnect() call is made to the SPS, with
   - uszReaderName == uszReaderName0200
   - unReaderNameLen == the length of uszReaderName
   - hCard == hCard0200.

Expected Results:

1. The call returns
   - the return code BSI_CARD_ABSENT or BSI_UNKNOWN_ERROR.

Note: There are no gscBsiUtilGetExtendedErrorText() assertions following assertion 2.6 because the card is presumably not connected.

3. gscBsiUtilDisconnect()

```
unsigned          long                    gscBsiUtilConnect(
    IN            UTILCardHandle                hCard
)
```

References:
1. GSC-IS 4.5.4.
2. GSC-IS E.4.3.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0300.

Assertion 3.1

Purpose: To test gscBsiUtilDisconnect() using valid parameters.

Scenario:
1. A gscBsiUtilDisconnect() call is made to the SPS, with
   - hCard == hCard0300.

Expected Results:
1. The call returns
   - the return code BSI_OK.
2. The card is disconnected.

Assertion 3.2

Purpose: To test gscBsiUtilDisconnect() using a bad handle.

Scenario:
1. A gscBsiUtilDisconnect() call is made to the SPS, with
   - hCard /= hCard0300.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE.
2. The card is still connected.

Assertion 9.3.2.1

Assertion 9.3.2.2

(See section 9.)

Assertion 3.3

Purpose: To test gscBsiUtilDisconnect() with a removed card.

Scenario:

1. The connected card is removed.

2. A gscBsiUtilDisconnect() call is made to the SPS, with
   - hCard == hCard0300.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

4. gscBsiUtilBeginTransaction()

```
unsigned   long       gscBsiUtilBeginTransaction(
    IN     int            hCard,
    IN     boolean        blType
)
```

References:
1. GSC-IS 4.5.6.
2. GSC-IS E.4.4.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0400.

Assertion 4.1

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with no existing transaction lock, using valid parameters.

Scenario:
1. There is no existing transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == TRUE.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_SPSSERVICE.

2. If the return code is BSI_OK, then a transaction is established with the smart card.

Assertion 4.2

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with no existing transaction lock, using valid parameters.

Scenario:
1. There is no existing transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == FALSE.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_SPSSERVICE.

2. If the return code is BSI_OK, then a transaction is established with the smart card.

Assertion 4.3

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with another application having an existing transaction lock, using valid parameters.

Scenario:
1. Another application has established a transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == FALSE.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED.

Assertion 9.4.3.1

Assertion 9.4.3.2

(See section 9.)

Assertion 4.4

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with another application having established a transaction lock, using valid parameters.

Scenario:
1. Another application has established a transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == TRUE.

Expected Results:
1. The call waits indefinitely.

Assertion 4.5

Purpose: To test gscBsiUtilBeginTransaction() as a non-blocking transaction call, with the current application having already established a transaction lock, using valid parameters.

Scenario:
1. The current application has established a transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == FALSE.

Expected Results:
1. The call returns

- the return code BSI_NOT_TRANSACTED.

Assertion 4.6

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with the current application having already established a transaction lock, using valid parameters.

Scenario:
1. The current application has established a transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard == hCard0400
   - blType == TRUE.

Expected Results:
1. The call returns
   - the return code BSI_NOT_TRANSACTED.

Assertion 9.4.6.1

Assertion 9.4.6.2

(See section 9.)

Assertion 4.7

Purpose: To test gscBsiUtilBeginTransaction() as a blocking transaction call, with no existing transaction lock, with a bad handle.

Scenario:
1. There is no existing transaction lock.

2. A gscBsiUtilBeginTransaction() call is made to the SPS, with
   - hCard /= hCard0400
   - blType == TRUE.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_SPSSERVICE.

2. No transaction is established with the smart card.

Assertion 9.4.7.1

Assertion 9.4.7.2

(See section 9.)

5. gscBsiUtilEndTransaction()

```
unsigned   long       gscBsiUtilEndTransaction(
           int            hCard
)
```

References:
1. GSC-IS 4.5.7.
2. GSC-IS E.4.5.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0500.

Assertion 5.1

Purpose: To test gscBsiUtilEndTransaction() with an existing transaction lock, using valid parameters.

Scenario:
1. The current application has established a transaction lock.

2. A gscBsiUtilEndTransaction() call is made to the SPS, with
   - hCard == hCard0500.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_SPSSERVICE.

2. If BSI_OK, the previously existing transaction lock is ended.

Assertion 5.2

Purpose: To test gscBsiUtilEndTransaction() with no existing transaction lock.

Scenario:
1. There is no existing transaction lock.

2. A gscBsiUtilEndTransaction() call is made to the SPS, with
   - hCard == hCard0500.

Expected Results:
1. The call returns
   - the return code BSI_NOT_TRANSACTED or the return code BSI_NO_SPSSERVICE.

Assertion 9.5.2.1

Assertion 9.5.2.2

(See section 9.)

Assertion 5.3

Purpose: To test gscBsiUtilEndTransaction() with a bad handle.

Scenario:

1. The current application has established a transaction lock.

2. A gscBsiUtilEndTransaction() call is made to the SPS, with
   - hCard /= hCard0500.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_SPSSERVICE.

2. If BSI_BAD_HANDLE, the previously existing transaction lock remains in effect.

Assertion 9.5.3.1

Assertion 9.5.3.2

(See section 9.)

6. gscBsiUtilGetVersion()

```
unsigned    long              gscBsiUtilGetVersion(
    INOUT   unsigned char *         uszVersion
    INOUT   unsigned long *         punVersionLen
)
```

References:
1. GSC-IS 4.5.8.
2. GSC-IS E.2.
3. GSC-IS E.4.6.

Assertion 6.1

Purpose: To test gscBsiUtilGetVersion() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A pointer punVersionLen0601 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilGetVersion() call is made to the SPS, with
   - uszVersion == NULL
   - punVersionLen == punVersionLen0601.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - punVersionLen0601 == a pointer to the required length of the uszVersion buffer.

b) Final Mode.

Scenario:
1. A string buffer uszVersion0601, with length pointed to by punVersionLen0601, is allocated.

2. A gscBsiUtilGetVersion() call is made to the SPS, with
   - uszVersion == uszVersion0601
   - punVersionLen == punVersionLen0601.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - uszVersion0601 == a string containing the BSI implementation version of the SPS.

Assertion 6.2

Purpose: To test gscBsiUtilGetVersion() using valid parameters (Discovery Method 2):

a) Discovery Mode--an insufficient buffer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A string uszVersion0602, not of sufficient length to hold the BSI version, is allocated.

2. A pointer punVersionLen0602 to an unsigned long integer whose value is the length of uszVersion0602, is allocated.

3. A gscBsiUtilGetVersion() call is made to the SPS, with
   - uszVersion == uszVersion0602
   - punVersionLen == punVersionLen0602.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER
   - punVersionLen0602 == a pointer to the required length of the uszVersion0602 buffer.

b) Final Mode.

Scenario:
1. uszVersion0602 is re-allocated to be of length pointed to by punVersionLen0602.

2. A gscBsiUtilGetVersion() call is made to the SPS, with
   - uszVersion == uszVersion0602
   - punVersionLen == punVersionLen0602.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - uszVersion0602 == a string containing the BSI implementation version of the SPS.

Assertion 6.3

Purpose: To test gscBsiUtilGetVersion() using a bad Version length (Discovery Method 2).

Scenario:
1. A string uszVersion0603 of sufficient length to hold the BSI version is allocated.

2. A pointer punVersionLen0603 to an unsigned long variable whose value is /= length of uszVersion0603, is allocated.

3. A gscBsiUtilGetVersion() call is made to the SPS, with
   - uszVersion == uszVersion0603
   - punVersionLen == punVersionLen0603.

Expected Results:

1. The call returns
    - the return code BSI_BAD_PARAM.

7. gscBsiUtilGetCardProperties()

```
unsigned    long                    gscBsiUtilGetCardProperties(
   IN       UTILCardHandle                hCard
   INOUT    unsigned char*                uszCCCUniqueID
   INOUT    unsigned long*                punCCCUniqueIDLen
   OUT      unsigned long*                punCardCapability
)
```

References:
1. GSC-IS 4.5.9.
2. GSC-IS E.2.
3. GSC-IS E.4.7.

Starting state for each Assertion:
1. A pointer punCardCapability0700 to an unsigned long variable is allocated.

Assertion 7.1

Purpose: To test gscBsiUtilGetCardProperties() using valid parameters (Discovery Method 1):
a) Discovery Mode—a NULL discovery pointer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0701.

2. A pointer punUniqueIDLen0701 to an unsigned long variable == 0 is allocated.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0701
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0701
   - punCardCapability == punCardCapability0700.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then punUniqueIDLen0701 == a pointer to the required length of the uszCCCUniqueID buffer.

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard0701.

2. A string buffer uszCCCUniqueID0701, with length pointed to by punUniqueIDLen0701, is allocated.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0701
   - uszCCCUniqueID == uszCCCUniqueID0701
   - punUniqueIDLen == punUniqueIDLen0701
   - punCardCapability == punCardCapability0700.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszCCCUniqueID0701 == a string containing the Card Capability Container ID
   - punUniqueIDLen0701 == a pointer to the length of uszCCCUniqueID0701
   - punCardCapability0700 == a pointer to one of the recognized bitwise masks identifying the provider of the connected card.

Assertion 7.2

Purpose: To test gscBsiUtilGetCardProperties() using valid parameters (Discovery Method 2):
a) Discovery Mode—an insufficient buffer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0702.

2. A string uszCCCUniqueID0702, not of sufficient length to hold the CCC ID, is allocated.

3. A pointer punUniqueIDLen0702 to an unsigned long variable whose value is the length of uszCCCUniqueID0702, is allocated.

4. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0702
   - uszCCCUniqueID == uszCCCUniqueID0702
   - punUniqueIDLen == punUniqueIDLen0702
   - punCardCapability == punCardCapability0700.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
   - punUniqueIDLen0702 == a pointer to the required length of uszCCCUniqueID0702.

Assertion 9.7.2.1

Assertion 9.5.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard0702.

2. uszCCCUniqueID0702 is re-allocated to be of length pointed to by punUniqueIDLen0702.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0702
   - uszCCCUniqueID == uszCCCUniqueID0702
   - punUniqueIDLen == punUniqueIDLen0702
   - punCardCapability == to punCardCapability0700.

Expected Results:

1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszCCCUniqueID0702 == a string containing the Card Capability Container ID
   - punUniqueIDLen0702 == a pointer to the length of uszCCCUniqueID0701
   - punCardCapability0700 == a pointer to one of the recognized bitwise masks identifying the provider of the connected card.

Assertion 7.3

Purpose: To test gscBsiUtilGetCardProperties() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0703.

2. A pointer punUniqueIDLen0703 to an unsigned long variable == 0 is allocated.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard /= hCard0703
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0703
   - punCardCapability == punCardCapability0700.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.7.3.1

Assertion 9.7.3.2

(See section 9.)

Assertion 7.4

Purpose: To test gscBsiUtilGetCardProperties() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0704.

2. A pointer punUniqueIDLen0704 to an unsigned long variable == 0 is allocated.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0704
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0704
   - punCardCapability == punCardCapability0700.

4. A string buffer uszCCCUniqueID0704, with length pointed to by punUniqueIDLen0704, is allocated.

5. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard /= hCard0704
   - uszCCCUniqueID == uszCCCUniqueID0704
   - punUniqueIDLen == punUniqueIDLen0704
   - punCardCapability == punCardCapability0700.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.7.4.1

Assertion 9.7.4.2

(See section 9.)

Assertion 7.5

Purpose: To test gscBsiUtilGetCardProperties() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0705.

2. A pointer punUniqueIDLen0705 to an unsigned long variable == 0 is allocated.

3. The connected card is removed from the reader.

4. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0705
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0705
   - punCardCapability == punCardCapability0700.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 7.6

Purpose: To test gscBsiUtilGetCardProperties() with a removed card (Discovery Method 1, Final Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0706.

2. A pointer punUniqueIDLen0706 to an unsigned long variable == 0 is allocated.

3. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0706
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0706
   - punCardCapability == punCardCapability0700.

4. A string buffer uszCCCUniqueID0706, with length pointed to by punUniqueIDLen0706, is allocated.

5. The connected card is removed from the reader.

6. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0706
   - uszCCCUniqueID == uszCCCUniqueID0706
   - punUniqueIDLen == punUniqueIDLen0706
   - punCardCapability == punCardCapability0700.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 7.7

Purpose: To test gscBsiUtilGetCardProperties() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0707.
3. A pointer punUniqueIDLen0707 to an unsigned long variable == 0 is allocated.
4. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0707
   - uszCCCUniqueID == NULL
   - punUniqueIDLen == punUniqueIDLen0707
   - punCardCapability == punCardCapability0700.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.7.7.1

Assertion 9.7.7.2

(See section 9.)

Assertion 7.8

Purpose: To test gscBsiUtilGetCardProperties() using a bad Unique ID length (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0708.
2. A string uszCCCUniqueID0708 of sufficient length to hold the CCC ID is allocated.
3. A pointer punUniqueIDLen0708 to an unsigned long variable punUniqueIDLen0708, whose value is /= the length of uszCCCUniqueID0708, is allocated.
4. A gscBsiUtilGetCardProperties() call is made to the SPS, with
   - hCard == hCard0708
   - uszCCCUniqueID == uszCCCUniqueID0708
   - punUniqueIDLen == punUniqueIDLen0708
   - punCardCapability == punCardCapability0700.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.7.8.1

Assertion 9.7.8.2

(See section 9.)

8. gscBsiUtilGetCardStatus()

```
unsigned    long              gscBsiUtilGetCardStatus(
    IN      UTILCardHandle    hCard
)
```

References:
1. GSC-IS 4.5.10.
2. GSC-IS E.4.8.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0800.

Assertion 8.1

Purpose: To test gscBsiUtilGetCardStatus() using valid parameters.

Scenario:
1. A gscBsiUtilGetCardStatus() call is made to the SPS, with
   - hCard == hCard0800.

Expected Results:
1. The call returns
   - the return code BSI_OK.

Assertion 8.2

Purpose: To test gscBsiUtilGetCardStatus() using a bad handle.

Scenario:
1. A gscBsiUtilGetCardStatus() call is made to the SPS, with
   - hCard /= hCard0800.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE.

Assertion 9.8.2.1

Assertion 9.8.2.2

(See section 9.)

Assertion 8.3

Purpose: To test gscBsiUtilGetCardStatus() with a removed card.

Scenario:
1. The card is removed.
2. A gscBsiUtilGetCardStatus() call is made to the SPS, with
   - hCard == hCard0800.

Expected Results:

1. The call returns
    - the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

9. gscBsiUtilGetExtendedErrorText()

```
unsigned    long                    gscBsiUtilGetExtendedErrorText(
    IN      UTILCardHandle                  hCard
    OUT     char *                          uszErrorText[255]
)
```

References:
1. GSC-IS 4.5.11.
2. GSC-IS E.4.9.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard0900.

2. A string uszErrorText0900 of length 255 is allocated.

Note: The function gscBsiUtilGetExtendedErrorText() can be called following many, but not all, of the other test assertion function calls. gscBsiUtilGetExtendedErrorText() requires, for example, that a card be connected to the reader. Since it is also unclear whether gscBsiUtilGetExtendedErrorText() needs to be capable of providing extended messages for the "successful" return codes BSI_OK or BSI_TERMINAL_AUTH, the function will not be tested for successful codes.

For those assertions X.Y for which calling gscBsiUtilGetExtendedErrorText() is meaningful, the gscBsiUtilGetExtendedErrorText() assertions are, in this document, specified immediately following the X.Y assertion as 9.X.Y.1 and 9.X.Y.2.

Assertion 9.X.Y.1

Purpose: To test gscBsiUtilGetExtendedErrorText(), using valid parameters, following each test corresponding to Assertion X.Y.

Scenario:
1. A test corresponding to Assertion X.Y has just been completed.

2. A gscBsiUtilGetExtendedErrorText() call is made to the SPS, with
   - hCard == hCard0900
   - uszErrorText == uszErrorText0900.

Expected Results:
1. The call returns
   - either the return code BSI_OK or the return code BSI_NO_TEXT_AVAILABLE
   - if BSI_OK is the code returned, then uszErrorText0900 == an extended error message
   - if BSI_NO_TEXT_AVAILABLE is the code returned, then uszErrorText0900== NULL.

Assertion 9.X.Y.2

Purpose: To test gscBsiUtilGetExtendedErrorText(), using a bad card handle and a good error text buffer, following each test corresponding to Assertion X.Y.

Scenario:

1. A test corresponding to Assertion X.Y has just been completed.

2. A gscBsiUtilGetExtendedErrorText() call is made to the SPS, with
   - hCard /= hCard0900
   - uszErrorText == uszErrorText0900.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE.

10. gscBsiUtilGetReaderList()

```
unsigned    long                    gscBsiUtilGetReaderList(
    INOUT   unsigned char *             uszReaderList
    INOUT   unsigned long *             punReaderListLen
)
```

References:
1. GSC-IS 4.5.12
2. GSC-IS E.2.
3. GSC-IS E.4.10.

Assertion 10.1

Purpose: To test gscBsiUtilGetReaderList() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A pointer punReaderListLen1001 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilGetReaderList() call is made to the SPS, with
   - uszReaderList == NULL
   - punReaderListLen == punReaderListLen1001.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - punReaderListLen1001 == a pointer to the required length of the uszReaderList buffer.

b) Final Mode.

Scenario:
1. A string buffer uszReaderList1001, with length pointed to by punReaderListLen1001, is allocated.

2. A gscBsiUtilGetReaderList() call is made to the SPS, with
   - uszReaderList == uszReaderList1001
   - punReaderListLen == punReaderListLen1001.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - uszReaderList1001 == a string containing the list of available readers.

Assertion 10.2

Purpose: To test gscBsiUtilGetReaderList() using valid parameters (Discovery Method 2):
   a) Discovery Mode--an insufficient buffer
   b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
   1. A string uszReaderList1002, not of sufficient length to hold the reader list, is allocated.

   2. A pointer punReaderListLen1002 to an unsigned long variable, whose value is the length of uszReaderList1002, is allocated.

   3. A gscBsiUtilGetReaderList() call is made to the SPS, with
      - uszReaderList == uszReaderList1002
      - punReaderListLen == punReaderListLen1002.

Expected Results:
   1. The call returns
      - the return code BSI_INSUFFICIENT_BUFFER
      - punReaderListLen1002 == a pointer to the required length of uszReaderList1002.

b) Final Mode.

Scenario:
   1. uszReaderList1002 is re-allocated to be of length pointed to by punReaderListLen1002.

   2. A gscBsiUtilGetReaderList() call is made to the SPS, with
      - uszReaderList == uszReaderList1002
      - punReaderListLen == punReaderListLen1002.

Expected Results:
   1. The call returns
      - the return code BSI_OK
      - uszReaderList1002 == a string containing the list of available readers.

Assertion 10.3

Purpose: To test gscBsiUtilGetReaderList() using a bad ReaderList length (Discovery Method 1).

Scenario:
   1. A string uszReaderList1003, of sufficient length to hold the reader list, is allocated.

   2. A pointer punReaderListLen1003 to an unsigned long variable, whose value is /= the length of uszReaderList1003, is allocated.

   3. A gscBsiUtilGetReaderList() call is made to the SPS, with
      - uszReaderList == uszReaderList1003
      - punReaderListLen == punReaderListLen1003.

Expected Results:

1. The call returns
    - the return code BSI_BAD_PARAM.

11. gscBsiUtilPassthru()

```
unsigned    long                    gscBsiUtilPassthru(
   IN       UTILCardHandle                hCard
   IN       unsigned char *               uszCardCommand
   IN       unsigned long                 unCardCommandLen
   INOUT    unsigned char *               uszCardResponse
   INOUT    unsigned long *               punCardResponseLen
)
```

References:
1. GSC-IS 4.5.13.
2. GSC-IS E.2.
3. GSC-IS E.4.11.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1100.

Assertion 11.1

Purpose: To test gscBsiUtilPassthru() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A pointer punCardResponseLen1101 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1101.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - punCardResponseLen1101 == a pointer to the required length of the uszCardResponse buffer.

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard1100.

2. A string buffer uszCardResponse1101, with length pointed to by punCardResponseLen1101, is allocated.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1101
   - punCardResponseLen == punCardResponseLen1101.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - uszCardResponse1101 == a string containing the APDU response from the connected card.

Assertion 11.2

Purpose: To test gscBsiUtilPassthru() using valid parameters (Discovery Method 2):
a) Discovery Mode--an insufficient buffer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A string uszCardResponse1102, not of sufficient length to hold the APDU response from the connected card, is allocated.

2. A pointer punCardResponseLen1102 to an unsigned long variable whose value is the length of uszCardResponse1102, is allocated.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1102
   - punCardResponseLen == punCardResponseLen1102.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER
   - punCardResponseLen1102 == a pointer to the required length of uszCardResponse1102.

Assertion 9.11.2.1

Assertion 9.11.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard1100.

2. uszCardResponse1102 is re-allocated to be of length pointed to by punCardResponseLen1102.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1102
   - punCardResponseLen == to punCardResponseLen1102.

Expected Results:
1. The call returns
   - the return code BSI_OK
   - uszCardResponse1102 == a string containing the APDU response from the connected card.

Assertion 11.3

Purpose: To test gscBsiUtilPassthru() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:
1. A pointer punCardResponseLen1103 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard /= hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1103.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE.

Assertion 9.11.3.1

Assertion 9.11.3.2

(See section 9.)

Assertion 11.4

Purpose: To test gscBsiUtilPassthru() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A pointer punCardResponseLen1104 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1104.

3. A string buffer uszCardResponse1104, with length pointed to by punCardResponseLen1104, is allocated.

4. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard /= hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1104
   - punCardResponseLen == punCardResponseLen1104.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE.

Assertion 9.11.4.1

Assertion 9.11.4.2

(See section 9.)

Assertion 11.5

Purpose: To test gscBsiUtilPassthru() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:
1. Another application has established a transaction lock.

2. A pointer punCardResponseLen1105 to an unsigned long variable == 0 is allocated.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1105.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED.

Assertion 9.11.5.1

Assertion 9.11.5.2

(See section 9.)

Assertion 11.6

Purpose: To test gscBsiUtilPassthru() using a bad CardCommand length (Discovery Method 1, Discovery Mode).

Scenario:

1. A pointer punCardResponseLen1106 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen /= the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1105.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM.

Assertion 9.11.6.1

Assertion 9.11.6.2

(See section 9.)

Assertion 11.7

Purpose: To test gscBsiUtilPassthru() using a bad CardCommand length (Discovery Method 1, Final Mode).

Scenario:

1. A pointer punCardResponseLen1107 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1107.

3. A string buffer uszCardResponse1107, with length pointed to by punCardResponseLen1107, is allocated.

4. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen /= the length of uszCardCommand
   - uszCardResponse == uszCardResponse1107
   - punCardResponseLen == punCardResponseLen1107.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM.

Assertion 9.11.7.1

Assertion 9.11.7.2

(See section 9.)

Assertion 11.8

Purpose: To test gscBsiUtilPassthru() using a bad CardResponse length.

Scenario:
1. A string uszCardResponse1108, of sufficient length to hold the APDU response from the connected card, is allocated.

2. A pointer punCardResponseLen1108 to an unsigned long variable, whose value is /= length of uszCardResponse1108, is allocated.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1108
   - punCardResponseLen == punCardResponseLen1108.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM.

Assertion 9.11.8.1

Assertion 9.11.8.2

(See section 9.)

Assertion 11.9

Purpose: To test gscBsiUtilPassthru() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A pointer punCardResponseLen1109 to an unsigned long variable == 0, is allocated.

2. The connected card is removed from the reader.

3. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1109.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 11.10

Purpose: To test gscBsiUtilPassthru() with a removed card (Discovery Method 1, Final Mode).

Scenario:
1. A pointer punCardResponseLen1110 to an unsigned long variable == 0 is allocated.

2. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == NULL
   - punCardResponseLen == punCardResponseLen1110.

3. A string buffer uszCardResponse1110, with length pointed to by punCardResponseLen1110, is allocated.

4. The connected card is removed from the reader.

5. A gscBsiUtilPassthru() call is made to the SPS, with
   - hCard == hCard1100
   - uszCardCommand == a string representing the APDU to be sent to the connected card
   - unCardCommandLen == the length of uszCardCommand
   - uszCardResponse == uszCardResponse1110
   - punCardResponseLen == punCardResponseLen1110.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

12. gscBsiUtilReleaseContext()

```
unsigned    long                    gscBsiUtilReleaseContext(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
)
```

References:
1. GSC-IS 4.5.14.
2. GSC-IS E.4.12.

Starting state for each Assertion:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1200.

2. The card has a target container.

3. An authenticated session has been established with the target container.

Assertion 12.1

Purpose: To test gscBsiUtilReleaseContext() using valid parameters.

Scenario:
1. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard == hCard1200
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID.

Expected Results:
1. The call returns
   - the return code BSI_OK.

2. There is no longer an authenticated session established with the target container.

Assertion 12.2

Purpose: To test gscBsiUtilReleaseContext() using a bad handle.

Scenario:
1. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard /= hCard1200
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE.

2. There continues to be an authenticated session established with the target container on the connected card.

Assertion 9.12.2.1

Assertion 9.12.2.2

(See section 9.)

Assertion 12.3

Purpose: To test gscBsiUtilReleaseContext() using a bad AID value.

Scenario:

1. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard == hCard1200
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID.
2. There continues to be an authenticated session established with the target container on the connected card.

Assertion 9.12.3.1

Assertion 9.12.3.2

(See section 9.)

Assertion 12.4

Purpose: To test gscBsiUtilReleaseContext() using a bad AID length.

Scenario:

1. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard == hCard1200
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM.
2. There continues to be an authenticated session established with the target container on the connected card.

Assertion 9.12.4.1

Assertion 9.12.4.2

(See section 9.)

Assertion 12.5

Purpose: To test gscBsiUtilReleaseContext() with a removed card.

Scenario:

1. The connected card is removed from the reader.

2. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard == hCard1200
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID.

Expected Results:

1. The call returns
the return code BSI_CARD_REMOVED.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 12.6

Purpose: To test gscBsiUtilReleaseContext() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.

2. A gscBsiUtilReleaseContext() call is made to the SPS, with
   - hCard == hCard1200
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED.

Assertion 9.12.6.1

Assertion 9.12.6.2

(See section 9.)

13. gscBsiGcDataCreate()

```
unsigned    long                    gscBsiGcDataCreate(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    IN      GCtag                       ucTag
    IN      unsigned char *             uszValue
    IN      unsigned long               unValueLen
)
```

References:
1. GSC-IS 4.6.1.
2. GSC-IS E.5.1.

Assertion 13.1

Purpose: To test gscBsiGcDataCreate() using valid parameters.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1301
   - the card has a target container
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1301
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - the specified data value is stored, with the specified tag, in the target container.

3. No other changes are made to the container structure of the connected card.

Assertion 13.2

Purpose: To test gscBsiGcDataCreate() using a bad handle.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1302

- the card has a target container
- an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard /= hCard1302
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.2.1

Assertion 9.13.2.2

(See section 9.)

Assertion 13.3

Purpose: To test gscBsiGcDataCreate() using a bad AID value.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1303
   - the card has a target container
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1303
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:
1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.3.1

Assertion 9.13.3.2

(See section 9.)

Assertion 13.4

Purpose: To test gscBsiGcDataCreate() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.
2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1304
   - the card has a target container
   - an authenticated session has been established with the target container.
   -
3. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1304
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.
2. No changes are made to the container structure of the connected card.

Assertion 9.13.4.1

Assertion 9.13.4.2

(See section 9.)

Assertion 13.5

Purpose: To test gscBsiGcDataCreate() using a bad AID length.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1305
   - the card has a target container
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
    - hCard == hCard1305
    - uszAID == the AID value of the target container
    - unAIDLen /= the length of uszAID
    - ucTag == a char whose value is not the tag of a data item in the target container
    - uszValue == a string that can be accommodated by the target container
    - unValueLen == the length of uszValue.

Expected Results:
1. The call returns
    - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card

Assertion 9.13.5.1

Assertion 9.13.5.2

(See section 9.)

Assertion 13.6

Purpose: To test gscBsiGcDataCreate() using a bad data value length.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1306
    - the card has a target container
    - an authenticated session has been established with the target container
    - the target container has enough available space to accommodate uszvalue1300.

2. A gscBsiGcDataCreate() call is made to the SPS, with
    - hCard == hCard1306
    - uszAID == the AID value of the target container
    - unAIDLen == the length of uszAID
    - ucTag == a char whose value is not the tag of a data item in the target container
    - uszValue == a string that can be accommodated by the target container
    - unValueLen /= the length of uszValue.

Expected Results:
1. The call returns
    - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.6.1

Assertion 9.13.6.2

(See section 9.)

Assertion 13.7

Purpose: To test gscBsiGcDataCreate() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1307
   - the card has a target container
   - an authenticated session has been established with the target container.

2. The connected card is removed from the reader.

3. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1307
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 13.8

Purpose: To test gscBsiGcDataCreate() by not fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1308
   - the card has a target container
   - an authenticated session has not been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1308
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container

- uszValue == a string that can be accommodated by the target container
- unValueLen == the length of uszValue.

Expected Results:

1. The call returns the return code BSI_ACCESS_DENIED or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.8.1

Assertion 9.13.8.2

(See section 9.)

Assertion 13.9

Purpose: To test gscBsiGcDataCreate() using a too-large data value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1309
   - the card has a target container
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1309
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of a data item in the target container
   - uszValue == a string that is too large to be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:

1. The call returns
   - the return code BSI_NO_MORE_SPACE or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.9.1

Assertion 9.13.9.2

(See section 9.)

Assertion 13.10

Purpose: To test gscBsiGcDataCreate() using the tag of a data item that already exists.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1310
   - the card has a target container
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataCreate() call is made to the SPS, with
   - hCard == hCard1310
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of a data item in the target container
   - uszValue == a string that can be accommodated by the target container
   - unValueLen == the length of uszValue.

Expected Results:

1. The call returns
   - the return code BSI_TAG_EXISTS or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.13.10.1

Assertion 9.13.10.2

(See section 9.)

14. gscBsiGcDataDelete()

```
unsigned    long                    gscBsiGcDataDelete(
    IN      UTILCardHandle              hCard
    IN      unsigned char*              uszAID
    IN      unsigned long               unAIDLen
    IN      GCtag                       ucTag
)
```

References:
1. GSC-IS 4.6.2.
2. GSC-IS E.5.2.

Assertion 14.1

Purpose: To test gscBsiGcDataDelete() using valid parameters.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1401
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with
   - hCard == hCard1401
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then there is no longer a data item with the tag ucTag stored in the target container.

3. No other changes are made to the container structure of the connected card.

Assertion 14.2

Purpose: To test gscBsiGcDataDelete() using a bad handle.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1402
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with

- hCard /= hCard1402
- uszAID == the AID value of the target container
- unAIDLen == the length of uszAID
- ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.2.1

Assertion 9.14.2.2

(See section 9.)

Assertion 14.3

Purpose: To test gscBsiGcDataDelete() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1403
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

3. A gscBsiGcDataDelete() call is made to the SPS, with
   - hCard == hCard1403
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
   - the return code BSI_BAD_TAG or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.3.1

Assertion 9.14.3.2

(See section 9.)

Assertion 14.4

Purpose: To test gscBsiGcDataDelete() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1404
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with
   - hCard == hCard1404
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.4.1

Assertion 9.14.4.2

(See section 9.)

Assertion 14.5

Purpose: To test gscBsiGcDataDelete() using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1405
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with
   - hCard == hCard1405
   - uszAID == the AID value of the target container
   - unAIDLen == /= the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.5.1

Assertion 9.14.5.2

(See section 9.)

Assertion 14.6

Purpose: To test gscBsiGcDataDelete() using a bad tag.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1406
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with
   - hCard == hCard1406
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is not the tag of an existing data item in the target container.

Expected Results:
1. The call returns
   - the return code BSI_BAD_TAG or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.6.1

Assertion 9.14.6.2

(See section 9.)

Assertion 14.7

Purpose: To test gscBsiGcDataDelete() with a removed card.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1407
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The connected card is removed from the reader,

3. A gscBsiGcDataDelete() call is made to the SPS, with
    - hCard == hCard1407
    - uszAID == the AID value of the target container
    - unAIDLen == the length of uszAID
    - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
    - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 14.8

Purpose: To test gscBsiGcDataDelete() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1408
    - the card has a target container, which contains at least one data item
    - an authenticated session has not been established with the target container.

2. A gscBsiGcDataDelete() call is made to the SPS, with
    - hCard == hCard1408
    - uszAID == the AID value of the target container
    - unAIDLen == the length of uszAID
    - ucTag == a char whose value is the tag of an existing data item in the target container.

Expected Results:

1. The call returns
    - the return code BSI_ACCESS_DENIED or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.14.8.1

Assertion 9.14.8.2

(See section 9.)

15. gscBsiGcGetContainerProperties()

```
unsigned    long                    gscBsiGcGetContainerProperties(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    OUT     GCacr *                     strctGCacr
    OUT     GCContainerSize *           strctContainerSizes
    OUT     unsigned char *             containerVersion
)
```

The GCacr structure is
- BSIAcr strctCreateACR
- BSIAcr strctDeleteACR
- BSIAcr strctReadTagListACR
- BSIAcr strctReadValueACR
- BSIAcr strctUpdateValueACR

The BSIAcr structure is

unsigned long	unACRType
unsigned long	unKeyIDOrReference
unsigned long	unAuthID
unsigned long	unACRID

The GCContainerSize structure is

unsigned long	unMaxNbDataItems
unsigned long	unMaxValueStorageSize

References:
1. GSC-IS 4.6.3.
2. GSC-IS E.5.3.

Starting state for each Assertion:
1. There exists a GCAcr variable strctGCacr1500.
2. There exists a GCContainerSize variable strctContainerSizes1500.
3. There exists a string containerVersion1500.

Assertion 15.1

Purpose: To test gscBsiGcGetContainerProperties() using valid parameters.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1501
   - the card has a target container.
2. A gscBsiGcGetContainerProperties() call is made to the SPS, with
   - hCard == hCard1501
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctGCacr == strctGCacr1500
   - strctContainerSizes == strctContainerSizes1500
   - containerVersion == containerVersion1500.

Expected Results:

1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - the members of strctGCacr1500 are set to indicate access control conditions for all operations
   - for a Virtual Machine card, unMaxNbDataItems == the size of the T-Buffer, and unMaxValueStorageSize == the size of the V-Buffer. For file system cards that cannot calculate the size of the T- and V- buffers, unMaxNbDataItems == 0 and unMaxValueStorageSize == 0
   - containerVersion1500 == the version of the target container.

Assertion 15.2

Purpose: To test gscBsiGcGetContainerProperties() using a bad handle.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1502
   - the card has a target container.

2. A gscBsiGcGetContainerProperties() call is made to the SPS, with
   - hCard /= hCard1502
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctGCacr == strctGCacr1500
   - strctContainerSizes == strctContainerSizes1500
   - containerVersion == containerVersion1500.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.15.2.1

Assertion 9.15.2.2

(See section 9.)

Assertion 15.3

Purpose: To test gscBsiGcGetContainerProperties() with another application having established a transaction lock.

Scenario:

1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1503
   - the card has a target container.

3. A gscBsiGcGetContainerProperties() call is made to the SPS, with

- hCard == hCard1503
- uszAID == the AID value of the target container
- unAIDLen == the length of uszAID
- strctGCacr == strctGCacr1500
- strctContainerSizes == strctContainerSizes1500
- containerVersion == containerVersion1500.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.15.3.1

Assertion 9.15.3.2

(See section 9.)

Assertion 15.4

Purpose: To test gscBsiGcGetContainerProperties() using a bad AID value.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1504.

2. A gscBsiGcGetContainerProperties() call is made to the SPS, with
   - hCard == hCard1504
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - strctGCacr == strctGCacr1500
   - strctContainerSizes == strctContainerSizes1500
   - uszContainerVersion == containerVersion1500.

Expected Results:
1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.15.4.1

Assertion 9.15.4.2

(See section 9.)

Assertion 15.5

Purpose: To test gscBsiGcGetContainerProperties() using a bad parameter.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1505
   - the card has a target container.

2. A gscBsiGcGetContainerProperties() call is made to the SPS, with
   - hCard == hCard1505
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - strctGCacr == strctGCacr1500
   - strctContainerSizes == strctContainerSizes1500
   - uszContainerVersion == uszContainerVersion1500.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.15.5.1

Assertion 9.15.5.2

(See section 9.)

Assertion 15.6

Purpose: To test gscBsiGcGetContainerProperties() with a removed card.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1506
   - the card has a target container.

2. The connected card is removed from the reader.

3. A gscBsiGcGetContainerProperties() call is made to the SPS, with
   - hCard == hCard1506
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctGCacr == strctGCacr1500
   - strctContainerSizes == strctContainerSizes1500
   - uszContainerVersion == uszContainerVersion1500.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

16. gscBsiGcReadTagList()

```
unsigned          long                    gscBsiGcReadTagList(
    IN            UTILCardHandle                hCard
    IN            unsigned char *               uszAID
    IN            unsigned long                 unAIDLen
    INOUT         GCtag *                       TagArray
    INOUT         unsigned long *               punNbTags
)
```

References:
1. GSC-IS 4.6.4.
2. GSC-IS E.2.
3. GSC-IS E.5.4.

Assertion 16.1

Purpose: To test gscBsiGcReadTagList() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1601
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The unsigned long * punNbTags1601 == 0 is allocated.

3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1601
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1601.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - punNbTags1601 == a pointer to the number of tags which would be contained in a tag array after a gscBsiGcReadTagList() Final Mode call.

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard1601.

2. A GCtag array TagArray1601 with number of elements at least equal to that pointed to by punNbTags1601, is allocated.

3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1601
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == TagArray1601
   - punNbTags == punNbTags1601.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - TagArray1601 == an array containing the list of tags for the target container.

Assertion 16.2

Purpose: To test gscBsiGcReadTagList() using valid parameters (Discovery Method 2):
a) Discovery Mode--an insufficient buffer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1602
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A GCtag array TagArray1602, with not enough elements to hold the list of tags in the container, is allocated.

3. A pointer punNbTags1602 to an unsigned long variable, whose value is the number of tags in TagArray1602, is allocated.

4. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1602
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == TagArray1602
   - punNbTags == punNbTags1602.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then

- punNbTags1602 == a pointer to the number of tags which would be contained in a tag array after a gscBsiGcReadTagList() Final Mode call.

Assertion 9.16.2.1

Assertion 9.16.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard1602.

2. TagArray1602 is re-allocated with enough elements to contain the number of tags pointed to by punNbTags1602.

3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1602
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == TagArray1602
   - punNbTags == punNbTags1602.

Expected Results:

1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - TagArray1602 == an array containing the list of tags for the target container.

Assertion 16.3

Purpose: To test gscBsiGcReadTagList() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1603
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The unsigned long * punNbTags1603 == 0 is allocated.

3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard /= hCard1603
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL

- punNbTags == punNbTags1603.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.3.1

Assertion 9.16.3.2

(See section 9.)

Assertion 16.4

Purpose: To test gscBsiGcReadTagList() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1604
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.
2. The unsigned long * punNbTags1604 == 0 is allocated.
3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1604
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1604.
4. A GCtag array TagArray1604 with number of elements at least equal to that pointed to by punNbTags1604, is allocated.
5. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard /= hCard1604
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == TagArray1600
   - punNbTags == punNbTags1604.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.4.1

Assertion 9.16.4.2

(See section 9.)

Assertion 16.5

Purpose: To test gscBsiGcReadTagList() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:

1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1605
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

3. The unsigned long * punNbTags1605 == 0 is allocated.

4. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1605
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1605.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.5.1

Assertion 9.16.5.2

(See section 9.)

Assertion 16.6

Purpose: To test gscBsiGcReadTagList() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1606
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The unsigned long * punNbTags1606 == 0 is allocated.

3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1606

- uszAID == a string that does not contain the correct AID for any container on the connected card
- unAIDLen == the length of uszAID
- TagArray == NULL
- punNbTags == punNbTags1606.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.6.1

Assertion 9.16.6.2

(See section 9.)

Assertion 16.7

Purpose: To test gscBsiGcReadTagList() using a bad parameter (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1607
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.
2. The unsigned long * punNbTags1607 == 0 is allocated.
3. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1607
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1607.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.7.1

Assertion 9.16.7.2

(See section 9.)

Assertion 16.8

Purpose: To test gscBsiGcReadTagList() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1608
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The connected card is removed from the reader.

3. The unsigned long * punNbTags1608 == 0 is allocated.

4. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1608
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1608.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 16.9

Purpose: To test gscBsiGcReadTagList() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1609
   - the card has a target container
   - an authenticated session has not been established with the target container.

2. The unsigned long * punNbTags1609 == 0 is allocated.

4. A gscBsiGcReadTagList() call is made to the SPS, with
   - hCard == hCard1609
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - TagArray == NULL
   - punNbTags == punNbTags1609.

Expected Results:

1. The call returns
   - the return code BSI_ACCESS_DENIED or the return code BSI_NO_CARDSERVICE.

Assertion 9.16.9.1

Assertion 9.16.9.2

(See section 9.)

17. gscBsiGcReadValue()

```
unsigned          long                    gscBsiGcReadValue(
      IN          UTILCardHandle                hCard
      IN          unsigned char *               uszAID
      IN          unsigned long                 unAIDLen
      IN          GCtag                         ucTag
      INOUT       unsigned char *               uszValue
      INOUT       unsigned long *               punValueLen
)
```

References:
1. GSC-IS 4.6.4.
2. GSC-IS E.2.
3. GSC-IS E.5.5.

Assertion 17.1

Purpose: To test gscBsiGcReadValue() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1701
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1701 to an unsigned long variable == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1701
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1701.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - punValueLen1701 == a pointer to the required length of the uszValue buffer.

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard1701

2. A string buffer uszValue1701, with length pointed to by punValueLen1701, is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1701
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == uszValue1701
   - unValueLen == punValueLen1701.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszValue1701 == a string containing the value associated with the specified tag.

Assertion 17.2

Purpose: To test gscBsiGcReadValue() using valid parameters (Discovery Method 2):
a) Discovery Mode--an insufficient buffer
b) Final Mode—an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1702
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A string uszValue1702, not of sufficient length to hold the value associated with the tag uctag1702, is allocated.

3. A pointer punValueLen1702 to an unsigned long variable, whose value is the length of uszValue1702, is allocated.

4. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1702
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == uszValue1702
   - punValueLen == punValueLen1702.

Expected Results:

1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
   - punValueLen1702 == a pointer to the required length of uszValue1702.

Assertion 9.17.2.1

Assertion 9.17.2.2

(See section 9.)

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard1702.

2. uszValue1702 is re-allocated to be of length pointed to by punValueLen1702.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1702
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == uszValue1702
   - punValueLen == punValueLen1702.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszValue1702 == a string containing the value associated with the specified tag.

Assertion 17.3

Purpose: To test gscBsiGcReadValue() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1703
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1703 to an unsigned long variable == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard /= hCard1703
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1703.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.3.1

Assertion 9.17.3.2

(See section 9.)

Assertion 17.4

Purpose: To test gscBsiGcReadValue() using a bad handle (Discovery Method 1, Final Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1704
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1704 to an unsigned long variable == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1704
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == ucTag1704
   - uszValue == NULL
   - punValueLen == punValueLen1704.

4. A string buffer uszValue1704, with length punValueLen1704, is allocated.

5. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard /= hCard1704
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == uszValue1704

- punValueLen == punValueLen1704.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.4.1

Assertion 9.17.4.2

(See section 9.)

Assertion 17.5

Purpose: To test gscBsiGcReadValue() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:
1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1705
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

3. A pointer punValueLen1705 to an unsigned long variable == 0 is allocated.

4. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1705
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1705.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.5.1

Assertion 9.17.5.2

(See section 9.)

Assertion 17.6

Purpose: To test gscBsiGcReadValue() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1706
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1706 to an unsigned long variable == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1706
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1706.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.6.1

Assertion 9.17.6.2

(See section 9.)

Assertion 17.7

Purpose: To test gscBsiGcReadValue() using a bad parameter (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1707
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1707 == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1707
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container

- uszValue == NULL
- punValueLen == punValueLen1707.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.7.1

Assertion 9.17.7.2

(See section 9.)

Assertion 17.8

Purpose: To test gscBsiGcReadValue() using a bad tag (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1708
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.
2. A pointer punValueLen1708 to an unsigned long variable == 0 is allocated.
3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1708
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == an invalid tag for the target container
   - uszValue == NULL
   - punValueLen == punValueLen1708.

Expected Results:
1. The call returns
   - the return code BSI_BAD_TAG or the return code BSI_NO_CARDSERVICE.

Assertion 9.17.8.1

Assertion 9.17.8.2

(See section 9.)

Assertion 17.9

Purpose: To test gscBsiGcReadValue() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1709
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A pointer punValueLen1709 to an unsigned long variable == 0 is allocated.

3. The connected card is removed from the reader.

4. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1709
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1709.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 17.10

Purpose: To test gscBsiGcReadValue() without fulfilling the applicable ACR (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1710
   - the card has a target container
   - an authenticated session has not been established with the target container.

2. A pointer punValueLen1710 to an unsigned long variable == 0 is allocated.

3. A gscBsiGcReadValue() call is made to the SPS, with
   - hCard == hCard1710
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == NULL
   - punValueLen == punValueLen1710.

Expected Results:

1. The call returns
   - the return code BSI_ACCESS_DENIED.

Assertion 9.17.10.1

Assertion 9.17.10.2

(See section 9.)

18. gscBsiGcUpdateValue()

```
unsigned    long                    gscBsiGcUpdateValue(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    IN      GCtag                       ucTag
    IN      unsigned char *             uszValue
    IN      unsigned long               unValueLen
)
```

References:
1. GSC-IS 4.6.6.
2. GSC-IS E.5.6.

Assertion 18.1

Purpose: To test gscBsiGcUpdateValue() using valid parameters.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1801
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1801
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then the new data item is stored, with the specified tag, in the target container.

3. No other changes are made to the container structure of the connected card.

Assertion 18.2

Purpose: To test gscBsiGcUpdateValue() using a bad handle.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1802
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard /= hCard1802
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.2.1

Assertion 9.18.2.2

(See section 9.)

Assertion 18.3

Purpose: To test gscBsiGcUpdateValue() with another application having established a transaction block.

Scenario:
1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1803
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

3. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1803
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item

- unValueLen == the length of the new data item.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.3.1

Assertion 9.18.3.2

(See section 9.)

Assertion 18.4

Purpose: To test gscBsiGcUpdateValue() using a bad AID value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1804
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1804
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item..

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.4.1

Assertion 9.18.4.2

(See section 9.)

Assertion 18.5

Purpose: To test gscBsiGcUpdateValue() using a bad parameter.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1805
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1805
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item..

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.5.1

Assertion 9.18.5.2

(See section 9.)

Assertion 18.6

Purpose: To test gscBsiGcUpdateValue() using a bad data value length.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1806
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1806
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item

- unValueLen /= the length of the new data item..

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.6.1

Assertion 9.18.6.2

(See section 9.)

Assertion 18.7

Purpose: To test gscBsiGcUpdateValue() using a bad tag.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1807
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1807
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == an invalid tag for the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item..

Expected Results:

1. The call returns
   - the return code BSI_BAD_TAG or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.7.1

Assertion 9.18.7.2

(See section 9.)

Assertion 18.8

Purpose: To test gscBsiGcUpdateValue() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1808
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. The connected card is removed from the reader.

3. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1808
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item..

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 18.9

Purpose: To test gscBsiGcUpdateValue() without fulfilling the applicable ACR.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1809
   - the card has a target container
   - an authenticated session has not been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1809
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, but which could be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item.

Expected Results:

1. The call returns
   - the return code BSI_ACCESS_DENIED or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.9.1

Assertion 9.18.9.2

(See section 9.)

Assertion 18.10

Purpose: To test gscBsiGcUpdateValue() using a too-large data value.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1810
   - the card has a target container, which contains at least one data item
   - an authenticated session has been established with the target container.

2. A gscBsiGcUpdateValue() call is made to the SPS, with
   - hCard == hCard1810
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucTag == a char whose value is the tag of an existing data item in the target container
   - uszValue == a new data item which is /= to the existing data item identified by the specified tag, and which could not be accommodated by the target container if it replaced the existing data item
   - unValueLen == the length of the new data item.

Expected Results:

1. The call returns
   - the return code BSI_NO_MORE_SPACE or the return code BSI_NO_CARDSERVICE.

2. No changes are made to the container structure of the connected card.

Assertion 9.18.10.1

Assertion 9.18.10.2

(See section 9.)

19. gscBsiGetChallenge()

```
unsigned    long                    gscBsiGetChallenge(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    INOUT   unsigned char *             uszChallenge
    INOUT   unsigned long *             punChallengeLen
)
```

References:
1. GSC-IS 4.7.1.
2. GSC-IS E.2.
3. GSC-IS E.6.1.

Assertion 19.1

Purpose: To test gscBsiGetChallenge() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1901
   - the card has a target container.

2. A pointer punChallengeLen1901 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1901
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1901.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - punChallengeLen1901 == a pointer to the required length of the uszChallenge buffer.

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard1901.

2. A string buffer uszChallenge1901, with length pointed to by punChallengeLen1901, is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1901
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == uszChallenge1901
   - punChallengeLen == punChallengeLen1901.

Expected Results:
1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszChallenge1901 == a string containing the random challenge returned from the connected card.

Assertion 19.2

Purpose: To test gscBsiGetChallenge() using valid parameters (Discovery Method 2):
a) Discovery Mode—an insufficient pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1902
   - the card has a target container.

2. A string uszChallenge1902, not of sufficient length to hold a random challenge, is allocated.

3. A pointer punChallengeLen1902 to an unsigned long variable, whose value is the length of uszChallenge1902, is allocated.

4. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1902
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == uszChallenge1902
   - punChallengeLen == punChallengeLen1902.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
   - punChallengeLen1902 == a pointer to the required length of the uszChallenge buffer.

Assertion 9.19.2.1

Assertion 9.19.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard1902.

2. uszChalleng1902 is re-allocated to be of length pointed to by punChallengeLen1902.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1902
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == uszChallenge1902
   - punChallengeLen == punChallengeLen1902.

Expected Results:

1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszChallenge1902 == a string containing the random challenge returned from the connected card.

Assertion 19.3

Purpose: To test gscBsiGetChallenge() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1903
   - the card has a target container.

2. A pointer punChallengeLen1903 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard /= hCard1903
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1903.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.19.3.1

Assertion 9.19.3.2

(See section 9.)

Assertion 19.4

Purpose: To test gscBsiGetChallenge() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1904
   - the card has a target container.

2. A pointer punChallengeLen1904 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1904
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1904.

4. A string buffer uszChallenge1904, with length pointed to by punChallengeLen1904, is allocated.

5. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard /= hCard19004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == uszChallenge1904
   - punChallengeLen == punChallengeLen1904.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.19.4.1

Assertion 9.19.4.2

(See section 9.)

Assertion 19.5

Purpose: To test gscBsiGetChallenge() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:

1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1905

- the card has a target container.

3. A pointer punChallengeLen1905 to an unsigned long variable == 0 is allocated.

4. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1905
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1905.

Expected Results:
1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.19.5.1

Assertion 9.19.5.2

(See section 9.)

Assertion 19.6

Purpose: To test gscBsiGetChallenge() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1906
   - the card has a target container.

2. A pointer punChallengeLen1906 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1906
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1906.

Expected Results:
1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.19.6.1

Assertion 9.19.6.2

(See section 9.)

Assertion 19.7

Purpose: To test gscBsiGetChallenge() using a bad AID length (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1907
   - the card has a target container.

2. A pointer punChallengeLen1907 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1907
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1907.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.19.7.1

Assertion 9.19.7.2

(See section 9.)

Assertion 19.8

Purpose: To test gscBsiGetChallenge() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard1908
   - the card has a target container.

2. A pointer punChallengeLen1908 to an unsigned long variable == 0 is allocated.

3. The connected card is removed from the reader.

4. A gscBsiGetChallenge() call is made to the SPS, with
   - hCard == hCard1908
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszChallenge == NULL
   - punChallengeLen == punChallengeLen1908.

Expected Results:

1. The call returns
    - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

20. gscBsiSkiInternalAuthenticate()

```
unsigned    long                    gscBsiSkiInternalAuthenticate(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    IN      unsigned char               ucAlgoID
    IN      unsigned char *             uszChallenge
    IN      unsigned long               unChallengeLen
    INOUT   unsigned char *             uszCryptogram
    INOUT   unsigned long *             punCryptogramLen
)
```

References:
1. GSC-IS 4.7.2
2. GSC-IS E.2.
3. GSC-IS E.6.2.

Assertion 20.1

Purpose: To test gscBsiSkiInternalAuthenticate() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2001
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2001 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2001
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2001.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then

- punCryptogramLen2001 == a pointer to the required length of the uszCryptogram buffer.

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard2001.

2. A string buffer uszCryptogram2001, with length pointed to by punCryptogramLen2001, is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2001
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == uszCryptogram2001
   - punCryptogramLen == punCryptogramLen2001.

Expected Results:

1. The call returns:
   - the return code BSI_OK, the return code BSI_TERMINAL_AUTH, or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszCryptogram2001 == a string containing the cryptogram computed by the connected card.

Assertion 20.2

Purpose: To test gscBsiSkiInternalAuthenticate() using valid parameters (Discovery Method 2):
  a) Discovery Mode—an insufficient pointer
  b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2003
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A string uszCryptogram2002, not of sufficient length to hold a cryptogram computed by the connected card, is allocated.

3. A pointer punCryptogramLen2002 to an unsigned long, whose value is the length of uszCryptogram2002, is allocated.

4. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
    - hCard == hCard2002
    - uszAID == the AID value of the target container
    - unAIDLen == the length of uszAID
    - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
    - uszChallenge == the address of a string containing the challenge submitted to the connected card
    - unChallengeLen == the length of uszChallenge
    - uszCryptogram == uszCryptogram2002
    - punCryptogramLen == punCryptogramLen2002.

Expected Results:

1. The call returns
    - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
    - punCryptogramLen2002 == a pointer to the required length of the uszCryptogram buffer.

Assertion 9.20.2.1

Assertion 9.20.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard2002.

2. uszCryptogram2002 is re-allocated to be of length pointed to by punCryptogramLen2002.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
    - hCard == hCard2002
    - uszAID == the AID value of the target container
    - unAIDLen == the length of uszAID
    - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
    - uszChallenge == the address of a string containing the challenge submitted to the connected card
    - unChallengeLen == the length of uszChallenge
    - uszCryptogram == uszCryptogram2002
    - punCryptogramLen == punCryptogramLen2002.

Expected Results:

1. The call returns:
    - the return code BSI_OK, the return code BSI_TERMINAL_AUTH, or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then

- uszCryptogram == a string containing the cryptogram computed by the connected card

Assertion 20.3

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2003
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2003 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard /= hCard2003
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2003.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.3.1

Assertion 9.20.3.2

(See section 9.)

Assertion 20.4

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2004
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2004 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard20004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2004.

4. A string buffer uszCryptogram2004, with length pointed to by punCryptogramLen2004, is allocated.

5. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard /= hCard20004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == uszCryptogram2004
   - punCryptogramLen == punCryptogramLen2004.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.4.1

Assertion 9.20.4.2

(See section 9.)

Assertion 20.5

Purpose: To test gscBsiSkiInternalAuthenticate() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:
1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2005
   - the card has a target SKI provider module

- an authenticated session has been established with the target SKI provider module.

3. A pointer punCryptogramLen2005 to an unsigned long variable == 0 is allocated.

4. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2005
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2005.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.5.1

Assertion 9.20.5.2

(See section 9.)

Assertion 20.6

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2006
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2006 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2006
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card

- unChallengeLen == the length of uszChallenge
- uszCryptogram == NULL
- punCryptogramLen == punCryptogramLen2006.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.6.1

Assertion 9.20.6.2

(See section 9.)

Assertion 20.7

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad AID length (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2007
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.
2. A pointer punCryptogramLen2007 to an unsigned long variable == 0 is allocated.
3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2007
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2007.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.7.1

Assertion 9.20.7.2

(See section 9.)

Assertion 20.8

Purpose: To test gscBsiSkiInternalAuthenticate() using a parameter (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2008
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2008 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2008
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen /= the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2008.

Expected Results:

4. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.8.1

Assertion 9.20.8.2

(See section 9.)

Assertion 20.9

Purpose: To test gscBsiSkiInternalAuthenticate() using a bad cryptographic algorithm identifier (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2009
   - the card has a target SKI provider module
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2009 to an unsigned long variable == 0 is allocated.

3. A string ucAlgoID2009 that does not contains a valid cryptographic algorithm for the connected card to use to encrypt the challenge is allocated.

4. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2009
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a string that does not contains a valid cryptographic algorithm for the connected card to use to encrypt the challenge is allocated
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2009.

Expected Results:

1. The call returns
   - the return code BSI_BAD_ALGO_ID or the return code BSI_NO_CARDSERVICE.

Assertion 9.20.9.1

Assertion 9.20.9.2

(See section 9.)

Assertion 20.10

Purpose: To test gscBsiSkiInternalAuthenticate() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2010
   - the card has a target SKI provider
   - an authenticated session has been established with the target SKI provider module.

2. A pointer punCryptogramLen2010 to an unsigned long variable == 0 is allocated.

3. The connected card is removed from the reader.

4. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2010
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge

- uszCryptogram == NULL
- punCryptogramLen == punCryptogramLen2010.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 20.11

Purpose: To test gscBsiSkiInternalAuthenticate() without fulfilling the applicable ACR (Discovery Method1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2011
   - the card has a target SKI provider module
   - an authenticated session has not been established with the target SKI provider module.

2. A pointer punCryptogramLen2011 to an unsigned long variable == 0 is allocated.

3. A gscBsiSkiInternalAuthenticate() call is made to the SPS, with
   - hCard == hCard2011
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that the connected card must use to encrypt the challenge
   - uszChallenge == the address of a string containing the challenge submitted to the connected card
   - unChallengeLen == the length of uszChallenge
   - uszCryptogram == NULL
   - punCryptogramLen == punCryptogramLen2011.

Expected Results:

1. The call returns
   - the return code BSI_ACCESS_DENIED.

Assertion 9.20.11.1

Assertion 9.20.11.2

(See section 9.)

21. gscBsiPkiCompute()

```
unsigned    long                    gscBsiPkiCompute(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    IN      unsigned char               ucAlgoID
    IN      unsigned char *             uszMessage
    IN      unsigned long               unMessageLen
    INOUT   unsigned char *             uszResult
    INOUT   unsigned long *             punResultLen
)
```

References:
1. GSC-IS 4.7.3.
2. GSC-IS E.2.
3. GSC-IS E.6.3.

Assertion 21.1

Purpose: To test gscBsiPkiCompute() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2101
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2101 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2101
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2101.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - punResultLen2101 == a pointer to the required length of the uszResult buffer.

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard2100.

2. A string buffer uszResult2101, with length pointed to by punResultLen2101, is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2101
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == uszResult2101
   - punResultLen == punResultLen2101.

Expected Results:

1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszResult2101 == a string containing the signature generated by the connected card.

Assertion 21.2

Purpose: To test gscBsiPkiCompute() using valid parameters (Discovery Method 2):

a) Discovery Mode—an insufficient pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2102
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A string uszResult2102, not of sufficient length to hold the signature generated by the connected card, is allocated.

3. A pointer punResultLen2102 to an unsigned long variable, whose value is the length of uszResult2102, is allocated.

4. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2102
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID

- ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
- uszMessage == a string containing the hash of the message to be signed
- unMessageLen == the length of uszMessage
- uszResult == uszResult2102
- punResultLen == punResultLen2102.

Expected Results:

1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
   - punResultLen2102 == a pointer to the required length of the uszResult buffer.

Assertion 9.21.2.1

Assertion 9.21.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard2102.

2. uszResult2102 is re-allocated to be of length pointed to by punResultLen2102.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2102
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == uszResult2102
   - punResultLen == punResultLen2102.

Expected Results:

1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszResult == a string containing the signature generated by the connected card.

Assertion 21.3

Purpose: To test gscBsiPkiCompute() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2103
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2103 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard /= hCard2103
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2103.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.3.1

Assertion 9.21.3.2

(See section 9.)

Assertion 21.4

Purpose: To test gscBsiPkiCompute() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2104
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2104 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard21004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID

- ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
- uszMessage == a string containing the hash of the message to be signed
- unMessageLen == the length of uszMessage
- uszResult == NULL
- punResultLen == punResultLen2104.

4. A string buffer uszResult2104, with length pointed to by punResultLen2104, is allocated.

5. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard /= hCard21004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == uszResult2104
   - punResultLen == punResultLen2104.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.4.1

Assertion 9.21.4.2

(See section 9.)

Assertion 21.5

Purpose: To test gscBsiPkiCompute() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:
1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2105
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

3. A pointer punResultLen2105 to an unsigned long variable == 0 is allocated.

4. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2105
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID

- ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
- uszMessage == a string containing the hash of the message to be signed
- unMessageLen == the length of uszMessage
- uszResult == NULL
- punResultLen == punResultLen2105.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.5.1

Assertion 9.21.5.2

(See section 9.)

Assertion 21.6

Purpose: To test gscBsiPkiCompute() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2106
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2106 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2106
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2106.

Expected Results:

1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.6.1

Assertion 9.21.6.2

(See section 9.)

Assertion 21.7

Purpose: To test gscBsiPkiCompute() using a bad AID length (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2107
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2107 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2107
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2107.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.7.1

Assertion 9.21.7.2

(See section 9.)

Assertion 21.8

Purpose: To test gscBsiPkiCompute() using a bad message length parameter (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2108
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2108 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2108
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen /= the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2108.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.8.1

Assertion 9.21.8.2

(See section 9.)

Assertion 21.9

Purpose: To test gscBsiPkiCompute() using a bad cryptographic algorithm identifier (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2109
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punResultLen2109 to an unsigned long variable == 0 is allocated.

3. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2109
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == A string that contains an invalid cryptographic algorithm for the connected card to use to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2109.

Expected Results:

1. The call returns
   - the return code BSI_BAD_ALGO_ID or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.9.1

Assertion 9.21.9.2

(See section 9.)

Assertion 21.10

Purpose: To test gscBsiPkiCompute() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2110
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.
2. A pointer punResultLen2110 to an unsigned long variable == 0 is allocated.
3. The connected card is removed from the reader.
4. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2110
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2110.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 21.11

Purpose: To test gscBsiPkiCompute() without fulfilling the applicable ACR (Discovery Method1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2111
   - the card has a target PKI provider module
   - an authenticated session has not been established with the target PKI provider module.

2. A pointer punResultLen2111 to an unsigned long variable == 0 is allocated.

5. A gscBsiPkiCompute() call is made to the SPS, with
   - hCard == hCard2111
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - ucAlgoID == a character that identifies the cryptographic algorithm that will be used to generate the signature
   - uszMessage == a string containing the hash of the message to be signed
   - unMessageLen == the length of uszMessage
   - uszResult == NULL
   - punResultLen == punResultLen2111.

Expected Results:

1. The call returns
   - the return code BSI_ACCESS_DENIED or the return code BSI_NO_CARDSERVICE.

Assertion 9.21.11.1

Assertion 9.21.11.2

(See section 9.)

22. gscBsiGetCertificate()

```
unsigned    long                    gscBsiGetCertificate(
    IN      UTILCardHandle              hCard
    IN      unsigned char *             uszAID
    IN      unsigned long               unAIDLen
    INOUT   unsigned char *             uszCertificate
    INOUT   unsigned long *             punCertificateLen
)
```

References:
1. GSC-IS 4.7.4.
2. GSC-IS E.2.
3. GSC-IS E.6.4.

Assertion 22.1

Purpose: To test gscBsiGetCertificate() using valid parameters (Discovery Method 1):
a) Discovery Mode--NULL discovery pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2201
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punCertificateLen2201 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2201
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2201.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - punCertificateLen2201 == a pointer to the required length of the uszCertificate buffer.

b) Final Mode.

Scenario:
1. The card from a) is still in the reader, connected with handle hCard2201.

2. A string buffer uszCertificate2201, with length pointed to by punCertificateLen2201, is allocated.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2201
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == uszCertificate2201
   - punCertificateLen == punCertificateLen2201.

Expected Results:
1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszCertificate2201 == a string containing the certificate returned from the connected card.

Assertion 22.2

Purpose: To test gscBsiGetCertificate() using valid parameters (Discovery Method 2):
a) Discovery Mode—an insufficient pointer
b) Final Mode--an allocated buffer.

a) Discovery Mode.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2202
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A string uszCertificate2202, not of sufficient length to hold the certificate, is allocated.

3. A pointer punCertificateLen2202 to an unsigned long variable, whose value is the length of uszCertificate2202, is allocated.

4. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2202
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == uszCertificate2202
   - punCertificateLen == punCertificateLen2202.

Expected Results:
1. The call returns
   - the return code BSI_INSUFFICIENT_BUFFER or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_INSUFFICIENT_BUFFER, then
   - punCertificateLen2202 == a pointer to the required length of the uszCertificate buffer.

Assertion 9.22.2.1

Assertion 9.22.2.2

(See section 9.)

b) Final Mode.

Scenario:

1. The card from a) is still in the reader, connected with handle hCard2202.

2. uszCertificate2202 is re-allocated to be of length pointed to by punCertificateLen2202.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2202
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == uszCertificate2202
   - punCertificateLen == punCertificateLen2202.

Expected Results:

1. The call returns:
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - uszCertificate2202 == a string containing the certificate returned from the connected card.

Assertion 22.3

Purpose: To test gscBsiGetCertificate() using a bad handle (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2203
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punCertificateLen2203 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard /= hCard2203
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2203.

Expected Results:

1. The call returns

- the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.22.3.1

Assertion 9.22.3.2

(See section 9.)

Assertion 22.4

Purpose: To test gscBsiGetCertificate() using a bad handle (Discovery Method 1, Final Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2204
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punCertificateLen2204 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard22004
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2204.

4. A string buffer uszCertificate2204, with length pointed to by punCertificateLen2204, is allocated.

5. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard /= hCard2204
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == uszCertificate2204
   - punCertificateLen == punCertificateLen2204.

Expected Results:

1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.22.4.1

Assertion 9.22.4.2

(See section 9.)

Assertion 22.5

Purpose: To test gscBsiGetCertificate() with another application having established a transaction lock (Discovery Method 1, Discovery Mode).

Scenario:

1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2205
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

3. A pointer punCertificateLen2205 to an unsigned long variable == 0 is allocated.

4. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2205
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2205.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.22.5.1

Assertion 9.22.5.2

(See section 9.)

Assertion 22.6

Purpose: To test gscBsiGetCertificate() using a bad AID value (Discovery Method 1, Discovery Mode).

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2206
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.

2. A pointer punCertificateLen2206 to an unsigned long variable == 0 is allocated.

3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2206
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL

- punCertificateLen == punCertificateLen2206.

Expected Results:
1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.22.6.1

Assertion 9.22.6.2

(See section 9.)

Assertion 22.7

Purpose: To test gscBsiGetCertificate() using a bad AID length (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2207
   - the card has a target PKI provider module
   - an authenticated session has been established with the target PKI provider module.
2. A pointer punCertificateLen2207 to an unsigned long variable == 0 is allocated.
3. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2207
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2207.

Expected Results:
1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.22.7.1

Assertion 9.22.7.2

(See section 9.)

Assertion 22.8

Purpose: To test gscBsiGetCertificate() with a removed card (Discovery Method 1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2208

- the card has a target PKI provider module
- an authenticated session has been established with the target PKI provider module.

2. A pointer punCertificateLen2208 to an unsigned long variable == 0 is allocated.

3. The connected card is removed from the reader.

4. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2208
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2208.

Expected Results:
1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().

Assertion 22.9

Purpose: To test gscBsiGetCertificate() without fulfilling the applicable ACR (Discovery Method1, Discovery Mode).

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2209
   - the card has a target PKI provider module
   - an authenticated session has not been established with the target PKI provider module.

2. A pointer punCertificateLen2209 to an unsigned long variable == 0 is allocated.

5. A gscBsiGetCertificate() call is made to the SPS, with
   - hCard == hCard2209
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - uszCertificate == NULL
   - punCertificateLen == punCertificateLen2209.

Expected Results:
1. The call returns
   - the return code BSI_ACCESS_DENIED.

Assertion 9.22.9.1

Assertion 9.22.9.2

(See section 9.)

23. gscBsiGetCryptoProperties()

```
unsigned    long                    gscBsiGetCryptoProperties(
    IN      UTILCardHandle                  hCard
    IN      unsigned char *                 uszAID
    IN      unsigned long                   unAIDLen
    OUT     CRYPTOacr *                     strctCRYPTOacr
    OUT     unsigned long                   punKeyLen
)

The CRYPTOacr structure is
    BSIAcr   strctGetChallengeACR
    BSIAcr   strctInternalAuthenticateACR
    BSIAcr   strctPkiComputeACR
    BSIAcr   strctCreateACR
    BSIAcr   strctDeleteACR
    BSIAcr   strctReadTagListACR
    BSIAcr   strctReadValueACR
    BSIAcr   strctUpdateValueACR

The BSIAcr structure is
      unsigned long     unACRType
      unsigned long     unKeyIDOrReference
      unsigned long     unAuthID
      unsigned long     unACRID
```

References:
1. GSC-IS 4.7.5.
2. GSC-IS E.6.5.

Starting state for each Assertion:
1. There exists a CRYPTOacr variable strctCRYPTOacr2300.

2. There exists an unsigned long variable punKeyLen2300.

Assertion 23.1

Purpose: To test gscBsiGetCryptoProperties() using valid parameters.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2301
   - the card has a target PKI provider container.

2. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard == hCard2301
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:
1. The call returns
   - the return code BSI_OK or the return code BSI_NO_CARDSERVICE.

2. If the return code is BSI_OK, then
   - the members of strctCRYPTOacr2300 are set to indicate access control conditions for all operations.
   - punKeyLen2300 == the length of the private key managed by the PKI provider.

Assertion 23.2

Purpose: To test gscBsiGetCryptoProperties() using a bad handle.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2302
   - the card has a target PKI provider container.

2. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard /= hCard2302
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:
1. The call returns
   - the return code BSI_BAD_HANDLE or the return code BSI_NO_CARDSERVICE.

Assertion 9.23.2.1

Assertion 9.23.2.2

(See section 9.)

Assertion 23.3

Purpose: To test gscBsiGetCryptoProperties() with another application having established a transaction lock.

Scenario:
1. Another application has established a transaction lock.

2. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2303
   - the card has a target PKI provider container.

3. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard == hCard2303
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:

1. The call returns
   - the return code BSI_SC_LOCKED or the return code BSI_NO_CARDSERVICE.

Assertion 9.23.3.1

Assertion 9.23.3.2

(See section 9.)

Assertion 23.4

Purpose: To test gscBsiGetCryptoProperties() using a bad AID value.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2304
   - the card has a target PKI provider container.

2. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard == hCard2304
   - uszAID == a string that does not contain the correct AID for any container on the connected card
   - unAIDLen == the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:
1. The call returns
   - the return code BSI_BAD_AID or the return code BSI_NO_CARDSERVICE.

Assertion 9.23.4.1

Assertion 9.23.4.2

(See section 9.)

Assertion 23.5

Purpose: To test gscBsiGetCryptoProperties() using a bad AID length.

Scenario:
1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2305
   - the card has a target PKI provider container.

2. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard == hCard2305
   - uszAID == the AID value of the target container
   - unAIDLen /= the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:

1. The call returns
   - the return code BSI_BAD_PARAM or the return code BSI_NO_CARDSERVICE.

Assertion 9.23.5.1

Assertion 9.23.5.2

(See section 9.)

Assertion 23.6

Purpose: To test gscBsiGetCryptoProperties() with a removed card.

Scenario:

1. A card that claims conformance to the GSC-IS is in a reader, connected with handle hCard2306
   - the card has a target PKI provider container.
2. The connected card is removed from the reader.
3. A gscBsiGetCryptoProperties() call is made to the SPS, with
   - hCard == hCard2306
   - uszAID == the AID value of the target container
   - unAIDLen == the length of uszAID
   - strctCRYPTOacr == strctCRYPTOacr2300
   - punKeyLen == punKeyLen2300.

Expected Results:

1. The call returns
   - the return code BSI_CARD_REMOVED or the return code BSI_NO_CARDSERVICE.

Note: The state of a system following the removal of a connected card is considered to be implementor defined, and not subject to further testing, including the testing of GetExtendedErrorText().